PURPOSE

TRASHMAN is a machine language utility for the TRS-80 Models I and III. It can reduce BASIC's string compression time by 95% or more. TRASHMAN needs only 578 bytes of memory, plus 2 bytes for each active string. It can be used with all the major Operating Systems and with most other machine language programs. It is not intended for use with CP/M systems.

What's String Compression?

When a BASIC program changes a string (words, names, descriptions), it moves it to a new place in memory, and leaves a hole in the old place. Eventually, all available memory gets used up and BASIC has to push the strings together to free up some space. This takes time. Lots of time. The computer stops running for seconds or minutes, and you may even think it's "crashed". The keyboard won't work, and until all the strings have been collected, you just have to sit and wait. Then things run for a while, until string compression is needed again. And again.

If you're using your computer for business, that wastes your money. If you're using it personally, it wastes your time. However, as soon as you start using TRASHMAN, those delays almost disappear. The program is very easy to use, and the instructions below will show you exactly what to do.

What's The Catch?

If a BASIC program uses only a few strings, very little time is wasted in string compression, and TRASHMAN won't be helpful. But, if hundreds of strings, including large string arrays, are used, TRASHMAN is just what you need.

SAMPLE TIMING COMPARISONS

#	SECONDS DELAY		PERCENT
STRINGS	NORMAL	TRASHMAN	IMPROVEMENT
250	11.8	0.7	94
500	45.8	1.6	96.5
1000	179.6	3.5	98
2000	713.2	7.8	98.9

INSTALLATION

TRASHMAN is distributed on a Model I single-density diskette, a format that can be read by any Model I or Model III computer or operating system. The program is called "TM/CMD". If you have a two-drive Model I, just copy it from drive 1 to drive 0. If you have a one-drive Model I, insert the distribution diskette in drive 0, press the <RESET> button, and follow the prompts to copy it to a diskette of your own. (This won't work with TRSDOS 2.7DD, so copy it onto a single-density diskette first, then move it with TRSDOS 2.7DD.)

If you have a two-drive Model III, insert a non-write-protected copy of your operating system in drive 0 (it must have at least one granule of free space), insert the TRASHMAN diskette in drive 1, and:

```
DOSPLUS:   1. CONVERT :1
           2. COPY TM/CMD:1 :0
LDOS:         COPY TM/CMD:1 :0
MULTIDOS:     COPY TM/CMD:1 :0
NEWDOS:    1. PDRIVE 0,1,TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
           2. COPY TM/CMD:1 :0
TRSDOS:       CONVERT :1 :0
```

Notes to NEWDOS/80 users: 1) If drive 1 has 80 tracks, then 'TI=AL' must be used. 2) After copying the program, restore PDRIVE to it's normal setting for your configuration.

If you have a one-drive Model III, insert the distribution diskette in drive 0, press <RESET>, and follow the prompts to copy the program to your own TRSDOS operating system diskette. This one-drive copy facility only supports TRSDOS.

USE

There are only three steps to follow in using TRASHMAN:

1. Issue 'TM' as a DOS command and write down the address it displays. On a 48K machine with no other high-memory machine language routines, this address will be '-578'. On a 32K machine, it'll be '-16962'. If you're using other high-memory routines, be sure to issue 'TM' last. It will move itself directly below your other high-memory routines, update the 'HIMEM' address, and display an adjusted address for you.
2. Issue the BASIC command for your Operating System, and set memory size if using NEWDOS 2.1.
3. Tell TRASHMAN how much space to reserve for string compression. Normally, this is done by adding one line of code to the very beginning of your BASIC programs, but it also can be done directly from the keyboard. If you add the line of code to your BASIC programs and then save those programs, they will request space automatically thereafter.

Your BASIC programs must tell TRASHMAN how much space is needed for string compression. Two bytes are needed for each active string, so if a program used 200 strings, then 400 bytes are needed. There's no particular upper limit to this, other than the amount of available memory that must be divided between your program and the compression area. More information about this will be given later.

This is the line of code to add to your BASIC programs:

```
1 DEFUSR=-578 : X=USR(400) : IF X <> 0 THEN STOP
```

'-578' and '400' are just examples, and you can use different numbers if appropriate. For example, if the address you wrote down in step 1 wasn't '-578', then use the value you wrote down instead. The '400' should be replaced by the appropriate value for the number of strings in your program. If you aren't sure how many there are, it's safer to over-estimate a bit. If you give too large a number, there may not be enough space available, in which case the program will stop on that 'STOP' shown in line 1 above. In this case, reduce the space request and try again.

CHECKOUT

Just run the BASIC program. If you've become familiar with where it used to pause for string compression, you should notice that those delays are either completely gone, or else reduced to just a few seconds (as shown in the table at the beginning of these instructions). If this isn't the case, you probably didn't allocate enough space: remember that you must request two bytes for each active string, (strings that are allocated but "null" dont count), and that each element of a string array counts separately. Since BASIC begins its arrays at element zero, not one, the following creates 303 strings, not 200:

DIM A$(100,2)

TRASHMAN can't tell whether you've requested enough space until it actually scans for currently-active strings. If there isn't enough space, it lets the computer do normal, slow compression.

If you request too much space, one of two things will happen: there won't be enough space left for your program to run (in which case, you'll get an "OUT OF MEMORY" or "OUT OF STRING SPACE" message from BASIC); or the wasted space will cause string compression to occur more often than necessary. In this second case, everything will still run, and probably a lot faster than without TRASHMAN, but the program won't run as fast as it could. The solution in either case, of course, is to adjust the space request. If all of memory was already needed by a particular program, it may not be possible to use TRASHMAN with it unless the program size or memory requirements are reduced by re-programming.

OTHER CONSIDERATIONS

The explanation so far is sufficient for normal use of TRASHMAN. What follows is useful, but optional material.

Effect on Memory

In order to allocate new space, TRASHMAN must "CLEAR 50". This destroys all active variables. If the space allocation is performed right after BASIC is invoked, this makes no difference, but if TRASHMAN is asked to allocate space in the middle of the execution of a program, everything up to that point will be cleared away. Similarly, chaining programs while retaining active variables is incompatible with new requests for space in TRASHMAN.

The solution is very simple: just issue the maximum space requirement as soon as BASIC is started, and don't make any further requests until the next time the "BASIC" command is issued.

Using TRASHMAN With Several BASIC Programs

The space allocation request only has to be re-issued if it has to be changed. Once BASIC is active and TRASHMAN has allocated space, that space is maintained even when other programs are run. If the computer is <RESET> or control is returned to DOS, then the space request must be re-issued once BASIC is used again.

Changing The Space Allocation

To change the amount of reserved string compression space, just issue another request, using the same format as the one shown in the example line above. It isn't necessary to do this unless a later program needs more compression space, or needs more space for other things (in which case the compression space should be reduced).

Using Modified Programs Without TRASHMAN

The added line of code causes BASIC to pass control to a specified address in memory (-578 in these examples). If the TRASHMAN command wasn't issued in DOS before this is done, nothing will be at that address, and the BASIC program probably will "crash". One way to avoid this problem is to include the "TRASHMAN" command in the "AUTO" function, or in a "DO" or "CHAIN" sequence that is related to the BASIC programs that need TRASHMAN. Another method would be to not modify any programs in the first place, and to issue the "DEFUSR" and "X=USR" commands directly from the keyboard after issuing the "TRASHMAN" and "BASIC" commands. However, making everything automatic will probably be a lot more convenient.

Information Returned from TRASHMAN

Whenever a space request is accepted, TRASHMAN performs a "CLEAR 50" at the same time. Space is allocated just below "HIMEM", and the values in "HIMEM" for DOS and BASIC are updated. If there isn't enough space for the new request, nothing will change and the value retuned ("X" in the example line above) will be set to "1" or "2" (1=not enough space available; 2=HIMEM was changed, so additional space cannot be allocated).

If another program has changed the contents of "HIMEM", TRASHMAN won't allocate additional space thereafter, since it can't be sure that such an allocation is safe. That's why TRASHMAN should be the last command issued before BASIC is started.

Determining How Much Space is Already Allocated

If the space request value is "-1", the amount of space currently allocated will be returned, but "CLEAR 50" won't be issued:

```
1 DEFUSR=-578 : PRINT USR(-1)
```

Deactivating TRASHMAN

To temporarily suppress TRASHMAN, request zero space:

```
1 DEFUSR=-578 : X=USR(0)
```

Later on, another space request can be made to bring TRASHMAN back into play.

To completely remove TRASHMAN from memory, specify "-2":

```
1 DEFUSR=-578 : X=USR(-2)
```

When this is done, TRASHMAN will disengage itself from BASIC, and if possible will restore "HIMEM" to the value it had before the TRASHMAN command was issued in the first place. However, if "HIMEM" has been changed by another program after TRASHMAN started, then "HIMEM" will be left alone.